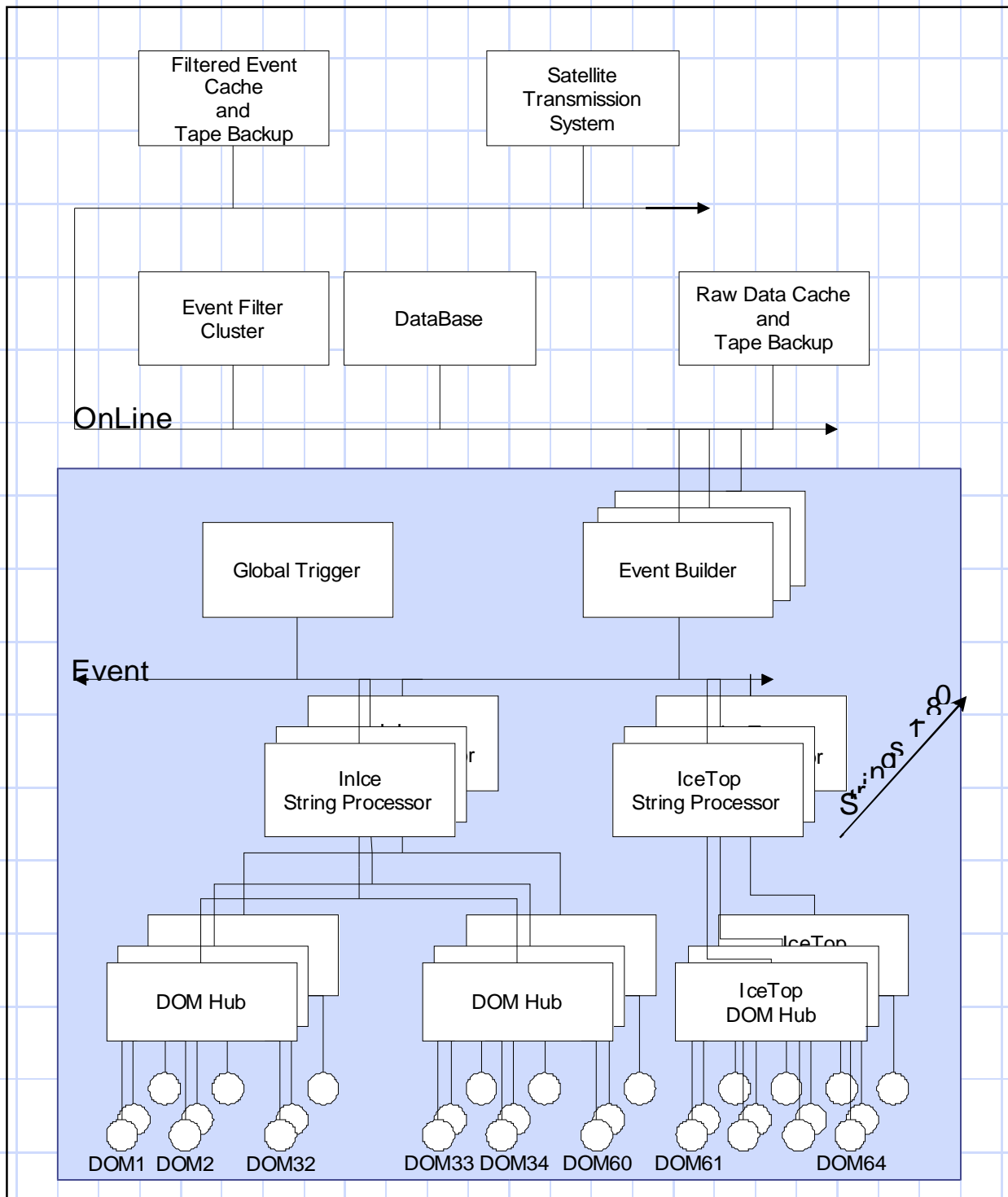


# IceCube DAQ: Simple Test Framework



# Test taxonomy

## ◆ Diagnostics

Is some part or sub system working?

Can we exercise it in a way that exposes the exact failure?

## ◆ Simple tests

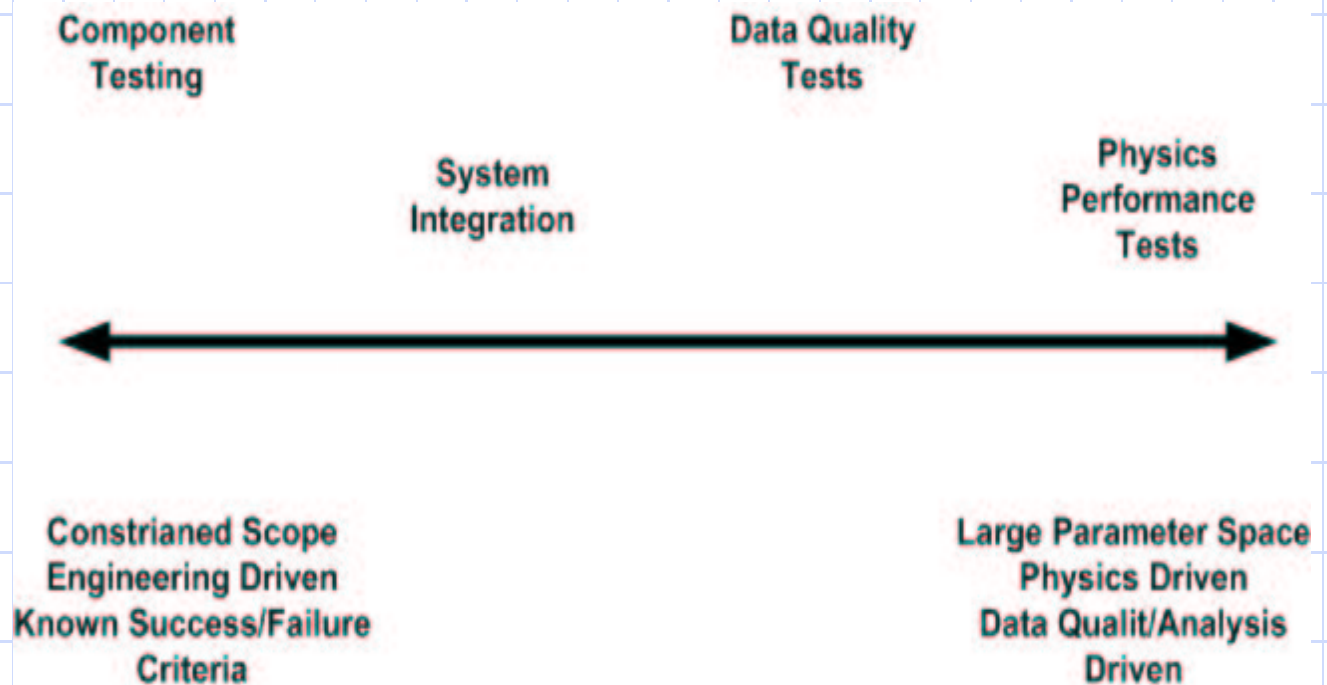
Does a part of sub system work as described by some criteria?

## ◆ Data analysis based tests

How well does some feature work?

How does a feature's performance vary with other system parameters?

# STF v.s. “physics/engineering” tests:



# Why do we need STF?

## ◆ Schedule

Only low level codes available when DOM MB first requires testing.

## ◆ Need to test basic functions.

Need to verify functioning of simple DOM MB blocks. (e.g. Fast ADC)

DOM application (DAQ) requires OS and DAQ FPGA design. These can influence behavior and will evolve over time.

## ◆ Repair, maintenance and QA

These functions need focused tests.

Same tests performed on bench and in ice.

## Why do we need STF? (contd.)

### ◆ Automated tests

- Efficiently run lengthy tests and log results.

- DAQ-based tests require skilled operators.

### ◆ Repeatability

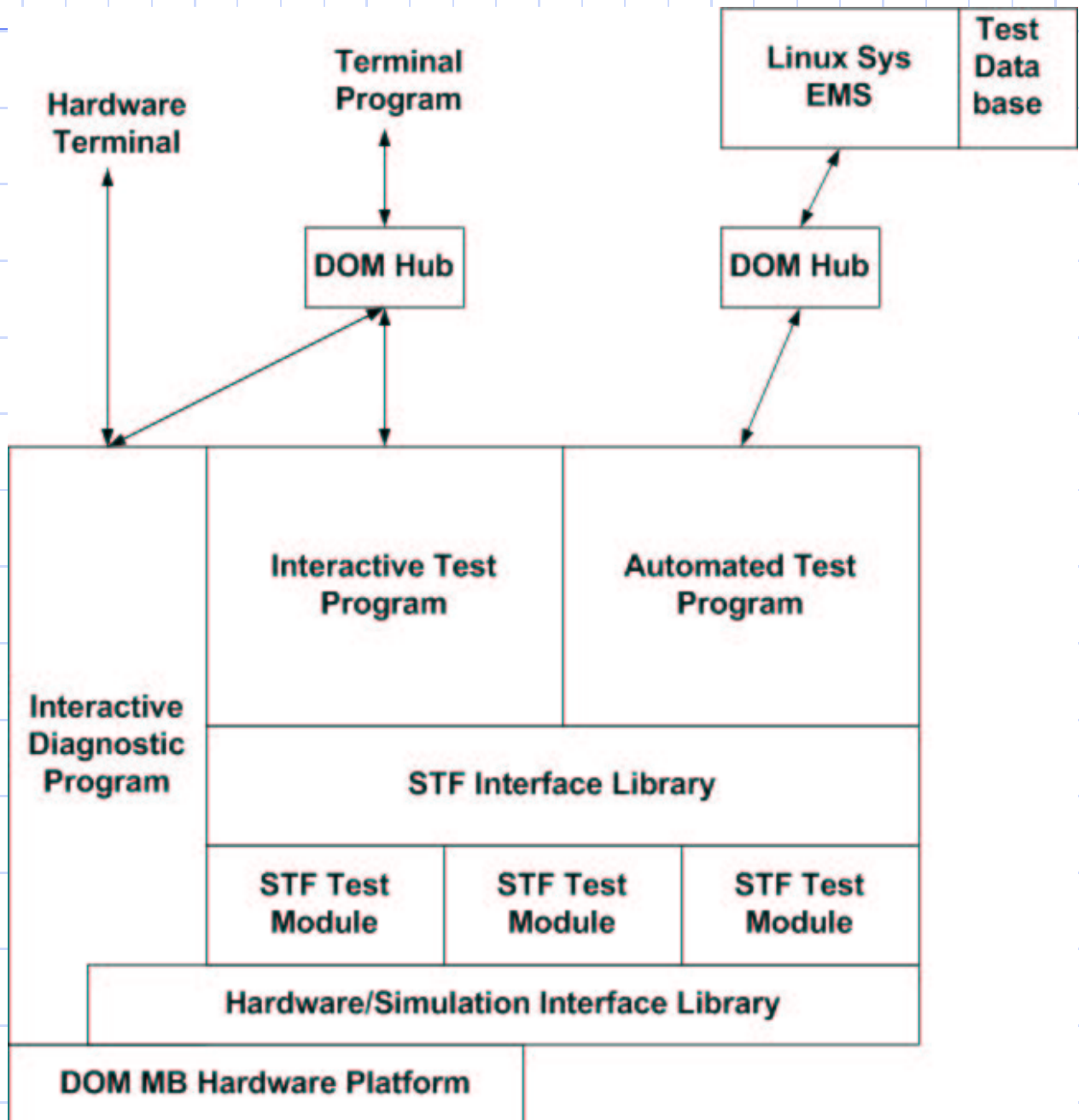
- Same tests/same conditions.

- Tracking of DOM performance over time.

## STF characteristics:

- ◆ Test code is DOM resident
- ◆ Known and usually simple pass/fail criteria
- ◆ Parameter driven
- ◆ Easily adapted to sequencing and automation

# STF Overview:



Software Test and Diagnostic Program Environment

## STF Module:

### ◆ Module entry points:

String getName();

String getVersion();

Void setParm(string name, string value);

Int performTest();

Int getCompletionStatus();

String getCompletionString();

String getParam(string name);

### ◆ Instance of executing a module maps well onto XML description.

Test DB can contain individual test description.

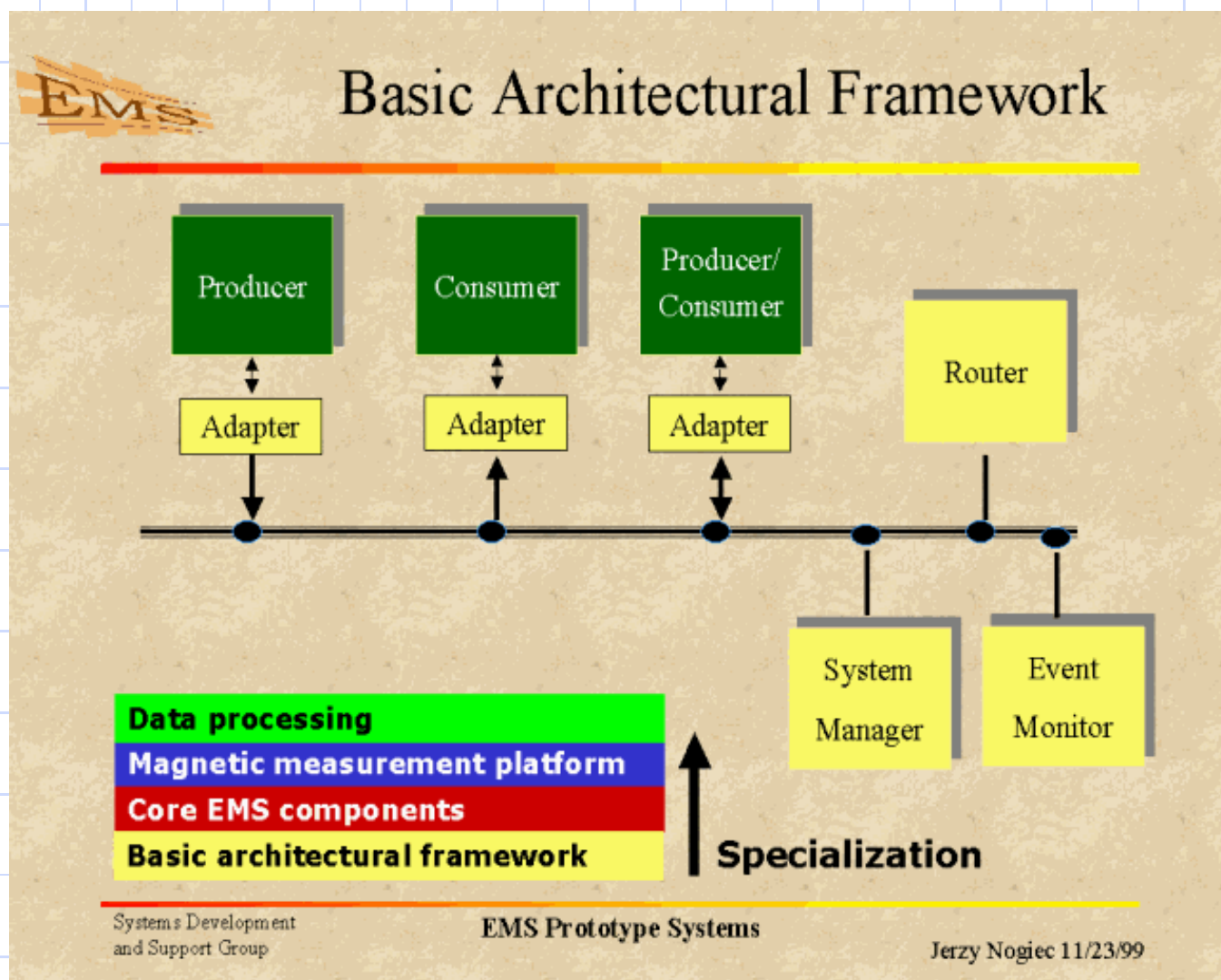
Sequence of tests easily driven from descriptions stored in test DB.

Test results stored in similar fashion.



# Production Test Environment: EMS Test Framework:

- ◆ Each test becomes a specific configuration of generic and specialized beans controlled through common interface.



# Production Test Environment: Test Examples:

## ◆ Level 0:

- ▮ Run memory and cpu diagnostics on “naked” DOM MB.
- ▮ Test PMT base on DOM MB.

## ◆ Level 1:

- ▮ Perform long term DOM↔DOM HUB communications test.
- ▮ Timing calibration tests between DOM and DOM HUB.

## ◆ Level 2:

- ▮ Verify network data transfer rates between DOM HUB and string processor.
- ▮ Etc.